البيانيات
المحاضرة 1

د. يسر السيد سليمان الأتاسي

10/13/2022

الهدف

- الإظهار البياني للأغراض
- طريقة العمل
- الدارات المستخدمة
- المواد المستعملة
- الظواهر الفيزيائية أو البيولوجية أو الكهربائية ...
- المحاكاة , التعديل , الحركة

Graphics Applications

- Entertainment: Cinema





Graphics Applications

- Entertainment: Games







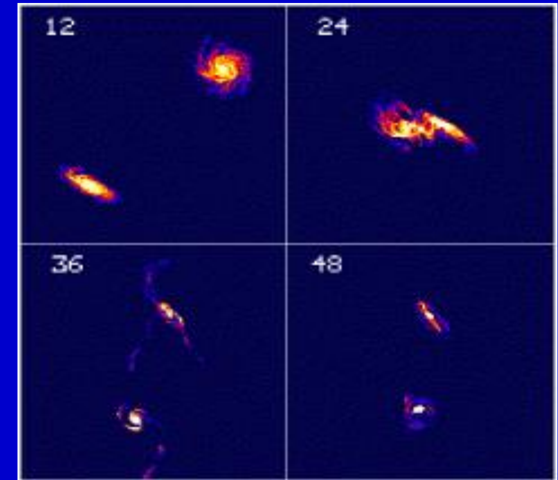
Graphics Applications

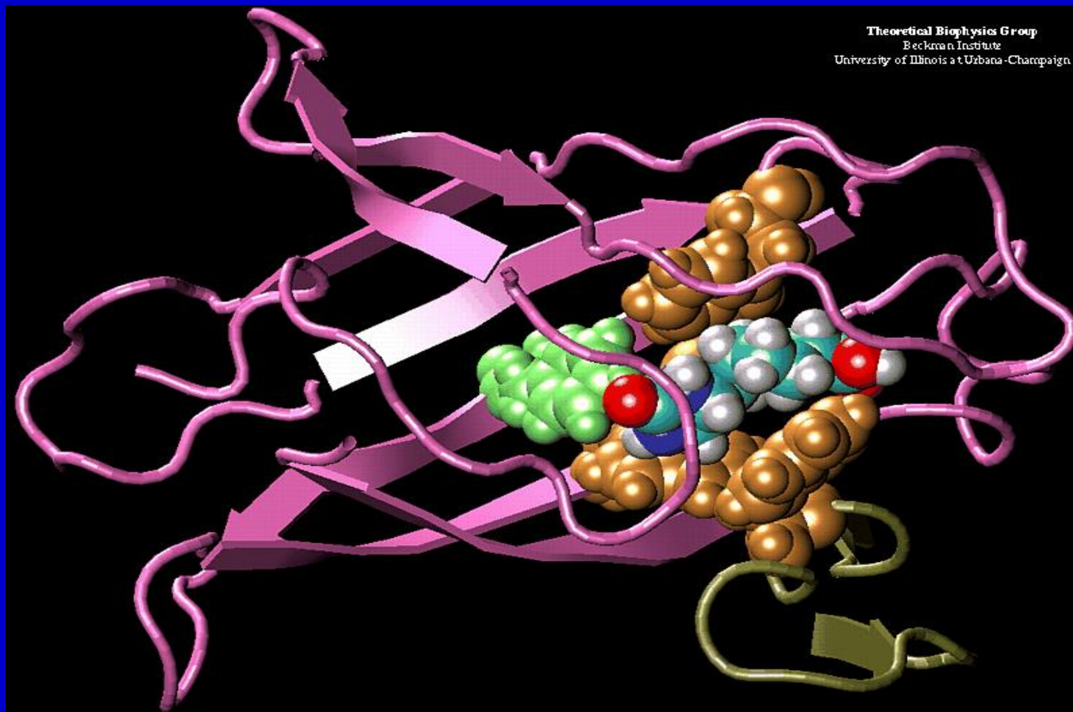
- Medical Visualization



Graphics Applications

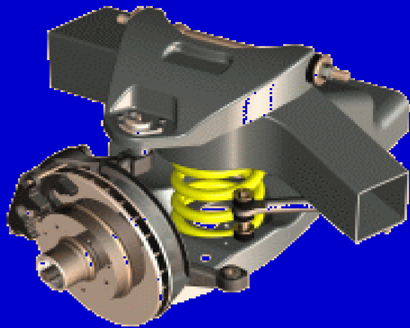
- Scientific Visualization

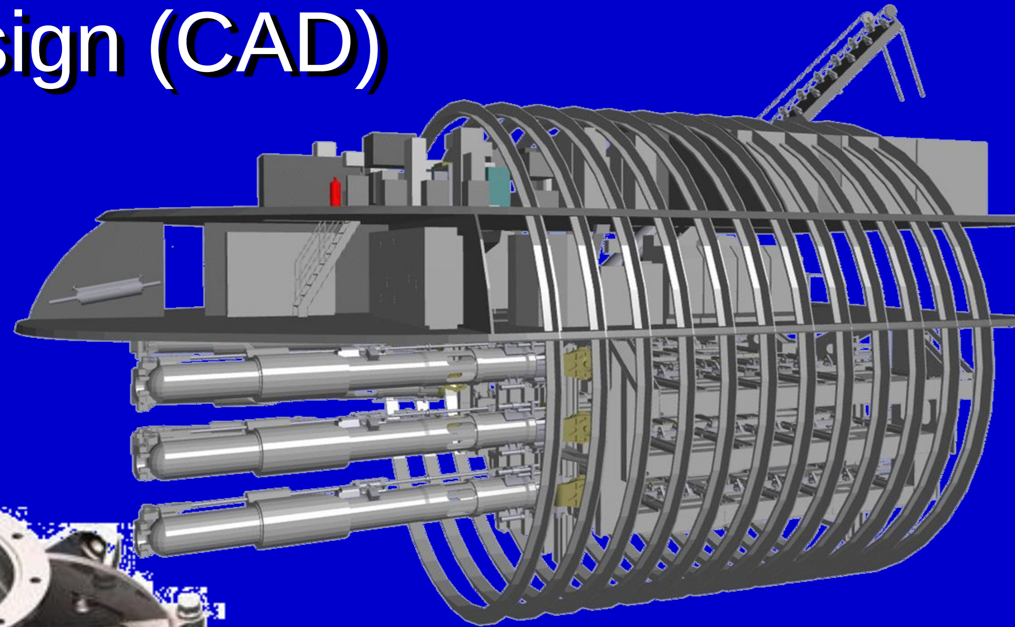


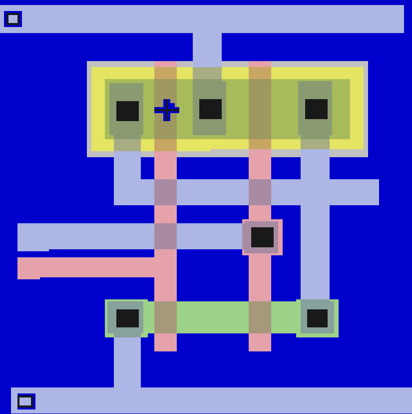


Graphics Applications

- Computer Aided Design (CAD)









المحتويات

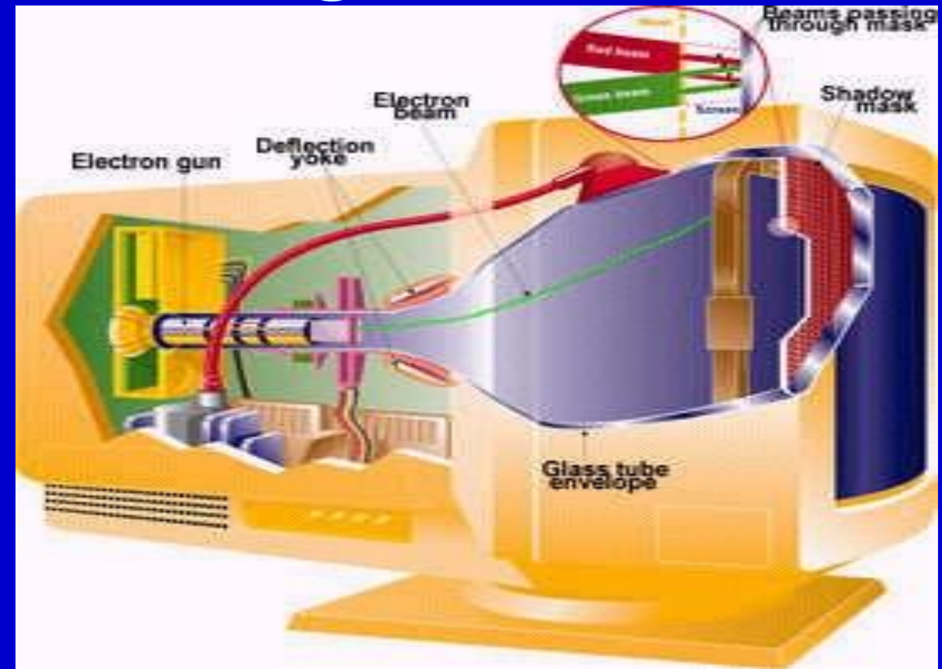
- مقدمة
- المفاهيم الأساسية في البعد الثنائي 2D (Rastering,transformations,clipping, filling,..)
- النمذجة الهندسية (الأجسام الصلبة – المنحنيات السطوح - الأغراض الكسرية)
- التحويلات ثلاثية الابعاد وفضاء الرؤية والاسقاطات وفضاء الجهاز
- الحركة (الأطر الأساسية- الحركة الإجرائية).
- الإظهار (الألوان –السطوح المرئية – الأشعة المطلقة)

Textbook

- Computer Graphics with OpenGL, Donald Hearn & Pauline Baker, Pearson, prentice Hall, third edition, 2004

Display Technologies

- Cathode Ray Tubes (CRTs)
  - Old display device
  - Evacuated glass bottle
  - Extremely high voltage
  - Heating element (filament)
  - Electrons pulled towards anode focusing cylinder



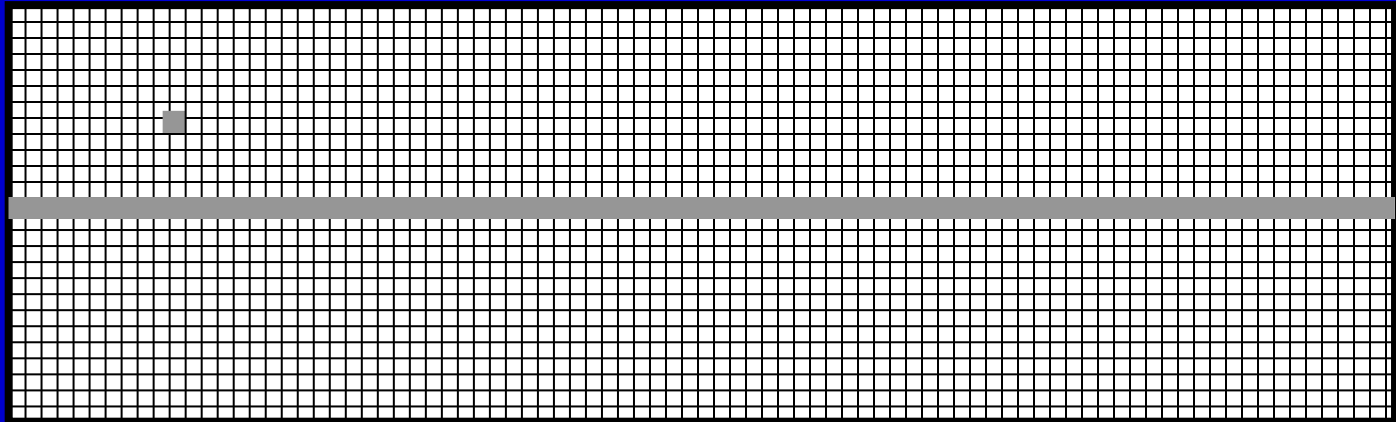
  - Vertical and horizontal deflection plates
  - Beam strikes phosphor coating on front of tube

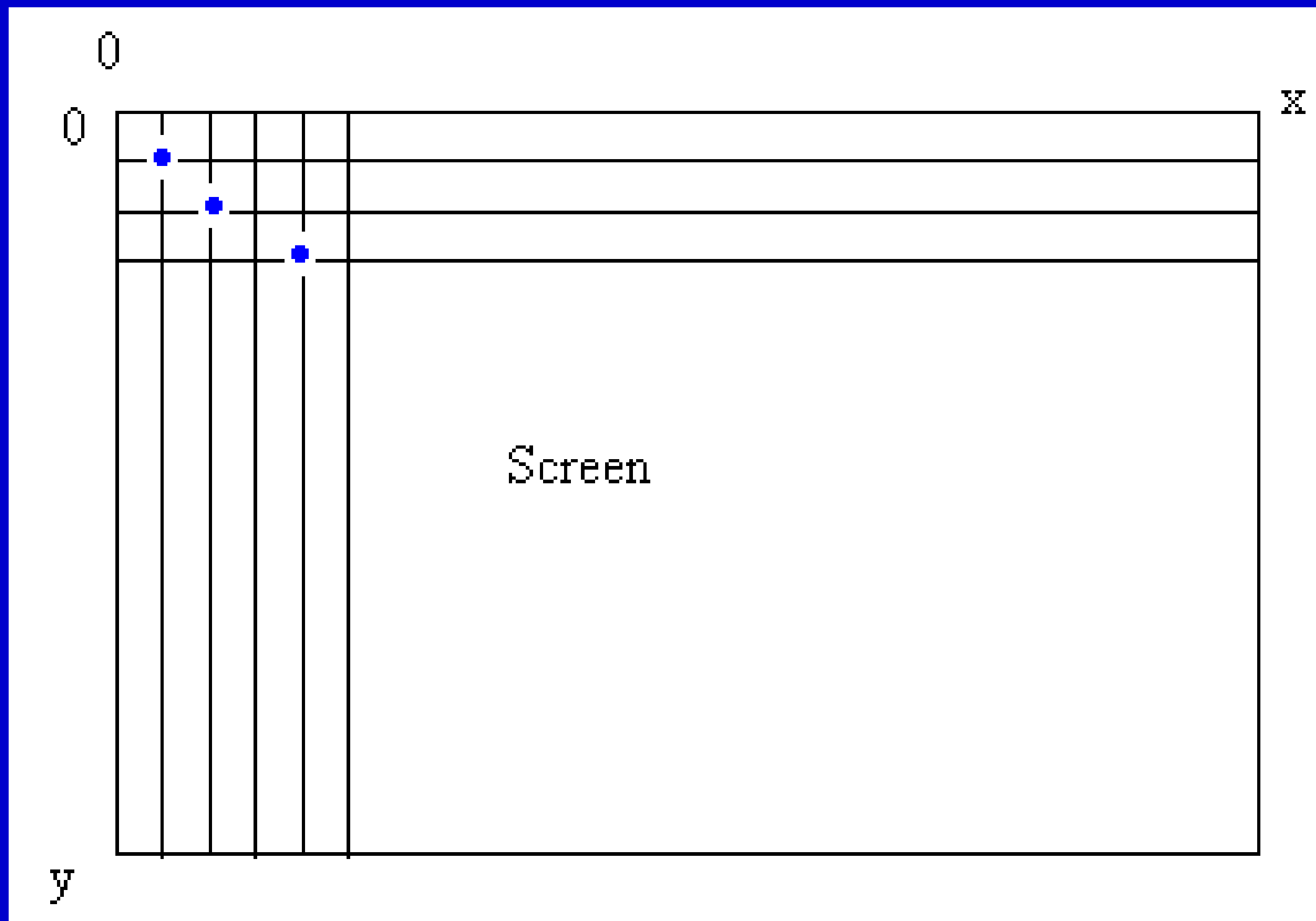
Display Technologies: CRTs

- Raster Displays
  - Black and white television: an oscilloscope with a fixed scan pattern: left to right, top to bottom
  - To paint the screen, computer needs to synchronize with the scanning pattern of raster
    - Solution: special memory to buffer image with scan-out synchronous to the raster. We call this the *framebuffer*.

Display Technologies: CRTs

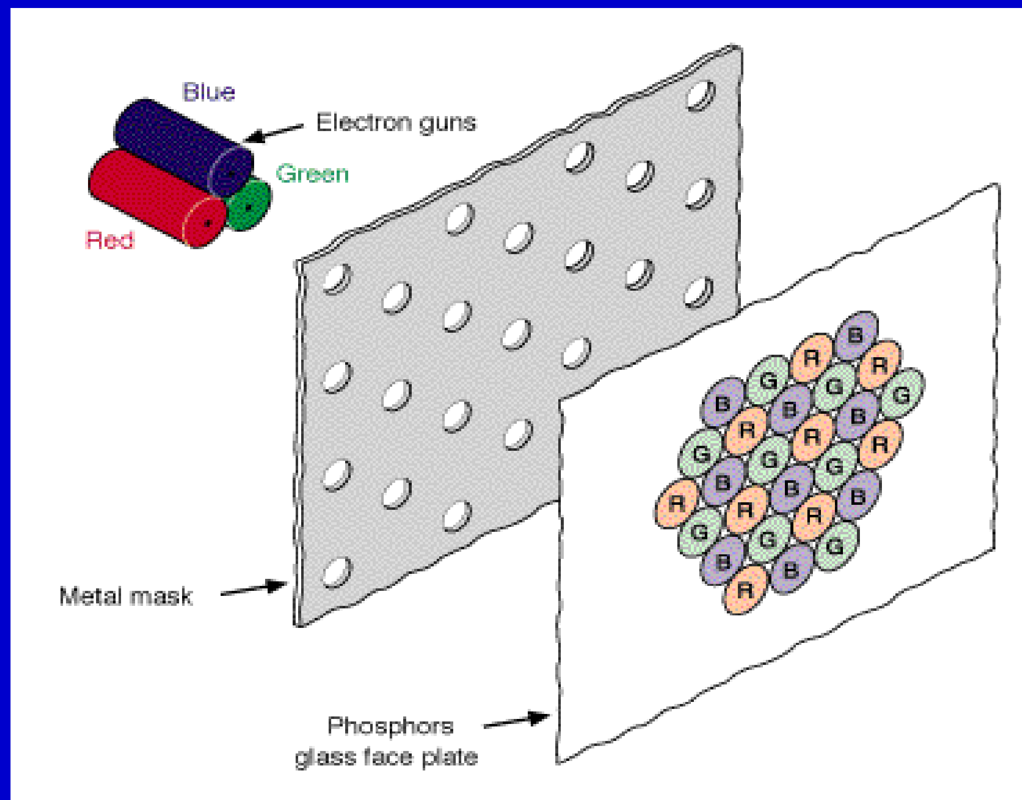
- Raster Displays
  - Raster: A rectangular array of points or dots
  - Pixel: One dot or picture element of the raster
  - Scan line: A row of pixels




0
0
x
Screen
y

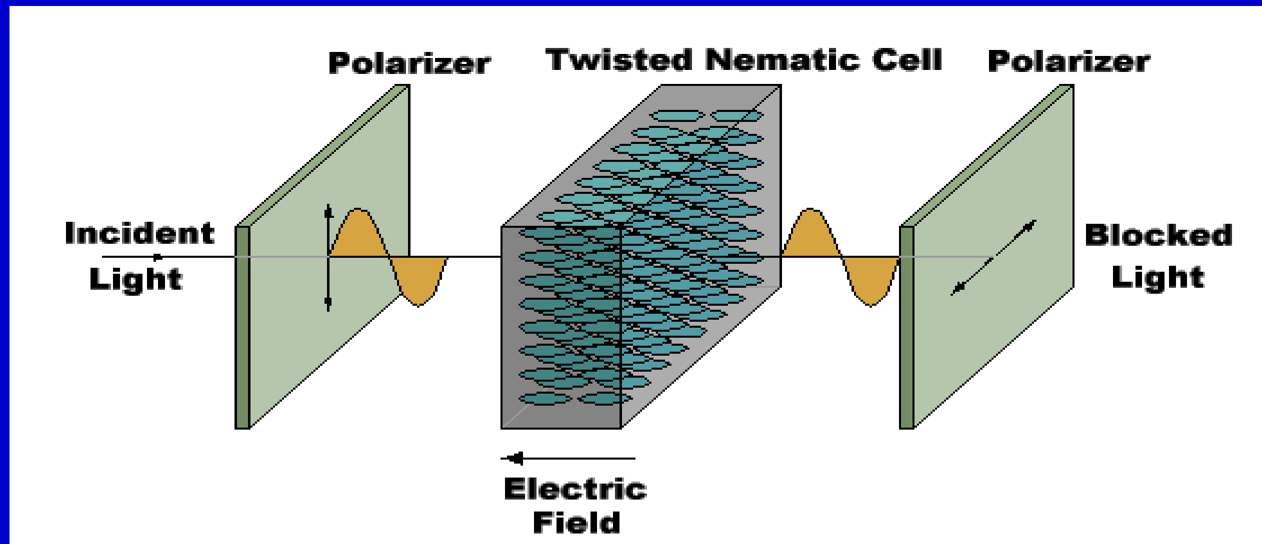
Display Technology: Color CRTs

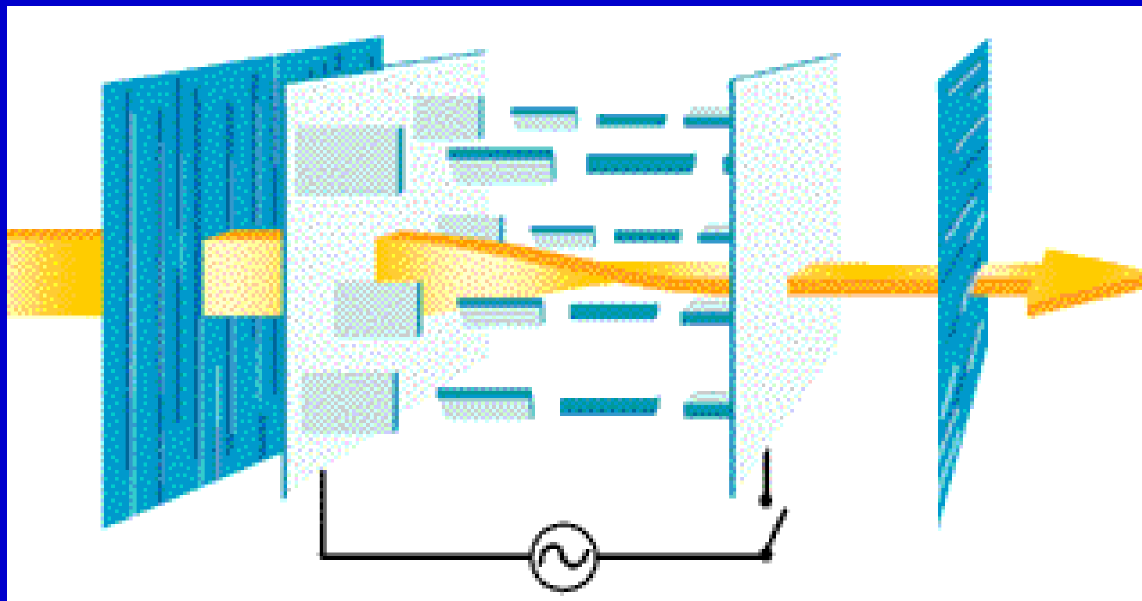
- Color CRTs have
  - Three electron guns
  - A metal *shadow mask* to differentiate the beams



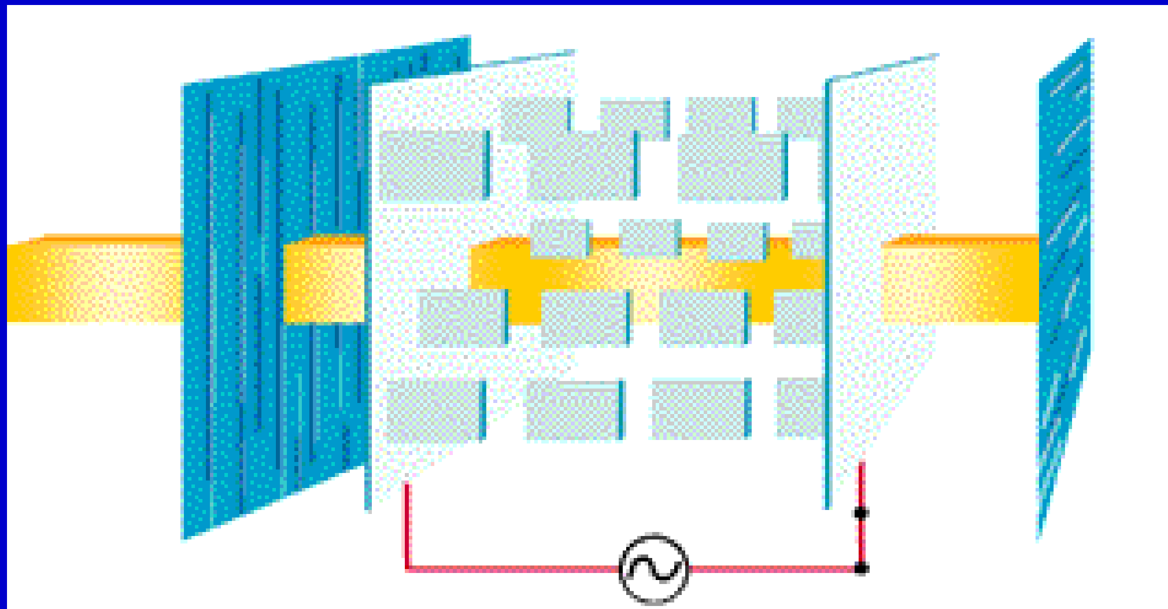
Display Technology: LCDs

- Liquid Crystal Displays (LCDs)
  - LCDs: organic molecules, naturally in crystalline state, that liquefy when excited by heat or E field
  - Crystalline state twists polarized light 90º





The liquid crystal status with no voltage applied



The liquid crystal with full control voltage applied

Mathematical Foundations

- review of some of the mathematical tools we'll use in **Geometry** (2D, 3D)
  - Trigonometry
  - Vector spaces
    - Points, vectors, and coordinates
  - Dot and cross products

Scalar Spaces

- Scalars: $\alpha, \beta, \ldots$
- Addition and multiplication (+ and $\bullet$) operations defined
- Scalar operations are
  - Associative: $\alpha + (\beta + \gamma) = (\alpha + \beta) + \gamma$
  - Commutative: $\alpha + \beta = \beta + \alpha \qquad \alpha \bullet \beta = \beta \bullet \alpha$
  - Distributive: $\alpha \bullet (\beta \bullet \gamma) = (\alpha \bullet \beta) \bullet \gamma$

    $\alpha \bullet (\beta + \gamma) = (\alpha \bullet \beta) + (\alpha \bullet \gamma)$

Scalar Spaces

- Additive Identity = 0
  - $\alpha + 0 = 0 + \alpha = \alpha$
- Multiplicative Identity = 1
  - $\alpha \bullet 1 = 1 \bullet \alpha = \alpha$
- Additive Inverse = $-\alpha$
  - $\alpha + (-\alpha) = 0$
- Multiplicative Inverse= α^{-1}
  - $\alpha \bullet \alpha^{-1} = 1$

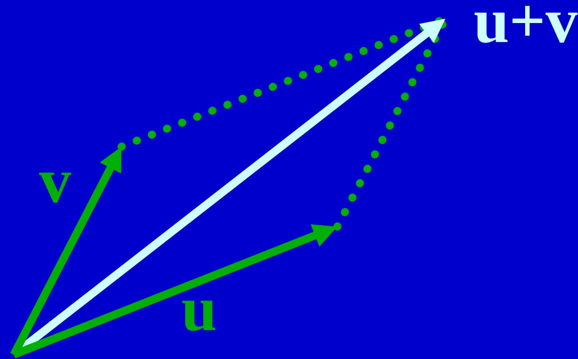
Vector Spaces

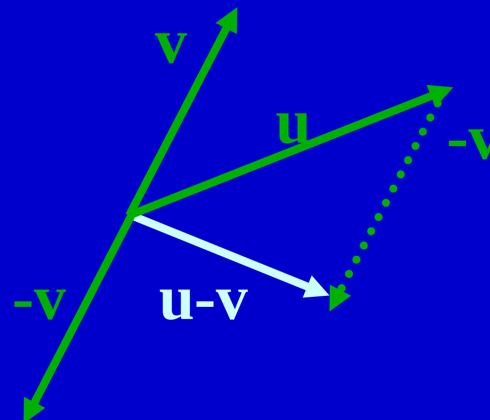
- Two types of elements:
  - Scalars (real numbers): $\alpha, \beta, \gamma, \delta, \ldots$
  - Vectors (n-tuples): $\mathbf{u}, \mathbf{v}, \mathbf{w}, \ldots$
- Operations:
  - Addition
  - Subtraction

Vector Addition/Subtraction

– operation **u + v**, with:
  - Identity **0**   **v** + **0** = **v**
  - Inverse **-**   **v** + (-**v**) = **0**

– Addition uses the “parallelogram rule”:



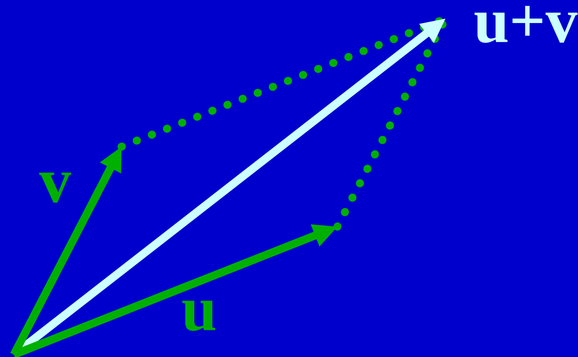


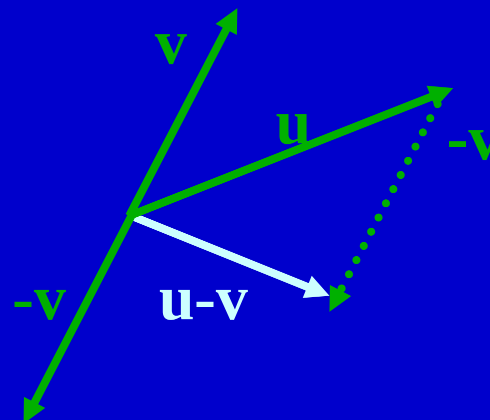
Vector Addition/Subtraction

– operation **u + v**, with:

- Identity **0** $\mathbf{v} + \mathbf{0} = \mathbf{v}$
- Inverse **-** $\mathbf{v} + (-\mathbf{v}) = \mathbf{0}$

– Addition uses the “parallelogram rule”:





Scalar Multiplication

– Scalar multiplication:
  - Distributive rule: $\alpha(\mathbf{u} + \mathbf{v}) = \alpha(\mathbf{u}) + \alpha(\mathbf{v})$

    $(\alpha + \beta)\mathbf{u} = \alpha\mathbf{u} + \beta\mathbf{u}$

– Scalar multiplication “streches” a vector, changing its length (*magnitude*) but not its direction

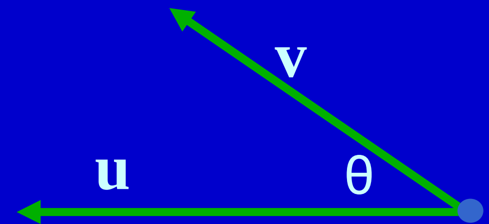
Dot Product

- The *dot product* or, more generally, *inner product* of two vectors is a scalar:

$$\mathbf{v}_1 \bullet \mathbf{v}_2 = \mathbf{x}_1\mathbf{x}_2 + \mathbf{y}_1\mathbf{y}_2 + \mathbf{z}_1\mathbf{z}_2 \qquad \text{(in 3D)}$$

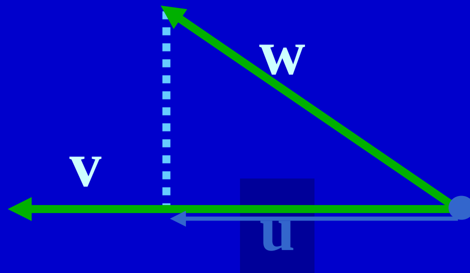
- Useful for many purposes
  - Computing the length (Euclidean Norm) of a vector: $length(\mathbf{v}) = ||\mathbf{v}|| = \mathrm{sqrt}(\mathbf{v} \bullet \mathbf{v})$
  - *Normalizing* a vector, making it unit-length: $\mathbf{v} = \mathbf{v} \,/\, ||\mathbf{v}||$
  - Computing the angle between two vectors:

    $\mathbf{u} \bullet \mathbf{v} = |\mathbf{u}|\ |\mathbf{v}|\ \cos(\theta)$
  - Checking two vectors for orthogonality
    - $\mathbf{u} \bullet \mathbf{v} = 0.0$



Dot Product

- *Projecting* one vector onto another
  - If **v** is a unit vector and we have another vector, **w**
  - We can project **w** perpendicularly onto **v**



$$
\begin{aligned}
\|u\| &= \|w\|\cos(\theta) \\
&= \|w\|\left(\frac{v \bullet w}{\|v\|\|w\|}\right) \\
&= v \bullet w
\end{aligned}
$$

  - And the result, **u**, has length **w • v**

Dot Product

- Is commutative
  - $u \cdot v = v \cdot u$
- Is distributive with respect to addition
  - $u \cdot (v + w) = u \cdot v + u \cdot w$

Cross Product

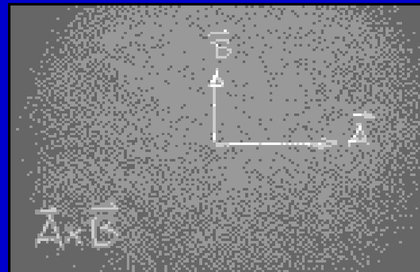
- The *cross product* or *vector product* of two vectors is a vector:

$$\mathbf{v}_1 \times \mathbf{v}_2 = \begin{bmatrix} y_1 z_2 - y_2 z_1 \\ -(x_1 z_2 - x_2 z_1) \\ x_1 y_2 - x_2 y_1 \end{bmatrix}$$

- The cross product of two vectors is orthogonal to both
- Right-hand rule dictates direction of cross product

Cross Product Right Hand Rule

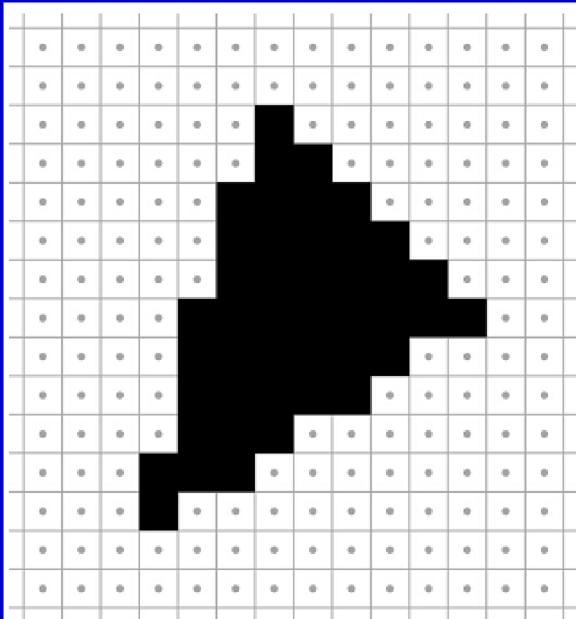
- See: http://www.phy.syr.edu/courses/video/RightHandRule/index2.html
- Orient your right hand such that your palm is at the beginning of A and your fingers point in the direction of A
- Twist your hand about the A-axis such that B extends perpendicularly from your palm
- As you curl your fingers to make a fist, your thumb will point in the direction of the cross product



Scan conversion (Raterization)

تحويل الشكل أو النموذج من شكله الهندسي إلى مصفوفة من البكسلات ووضعها في Frame buffer وتدعى بتكسير الشكل

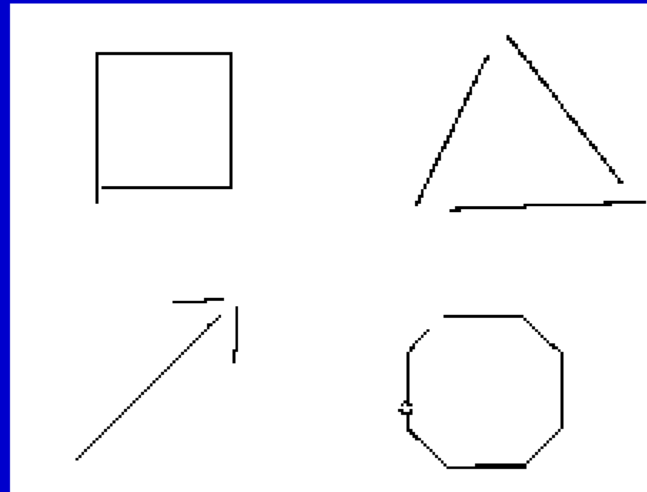
المكتبة البيانية SRGP (Simple Raster Graphics Package).



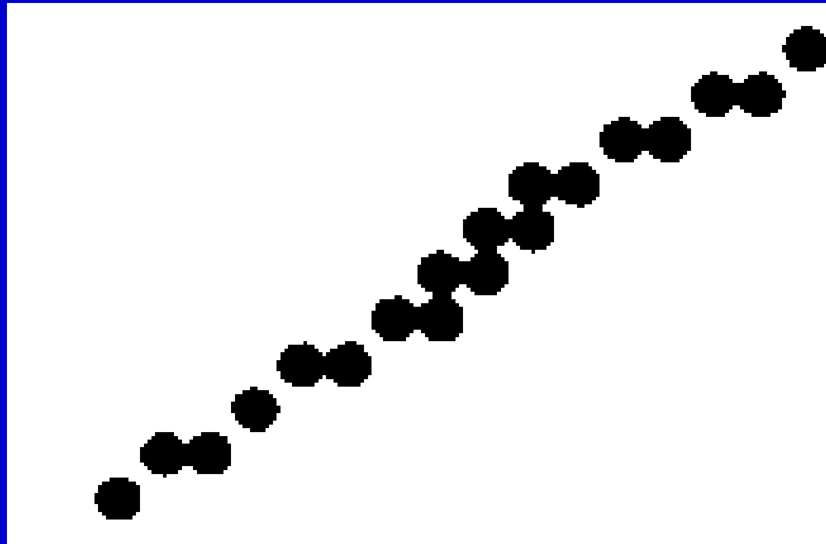
- المستقيم line- الدائرة circle .
- التعبئة Filling
- القطع clipping
- التحويلات ثنائية البعد.

خوارزميات رسم مستقيم

- يجب أن لا تظهر تكسيرات
- يجب أن تكون نهايات المستقيم دقيقة
- توزع كثافة نقاط المستقيم متساوية
- رسم المستقيم سريعاً



نهايات المستقيمات غير دقيقة



توزع كثافة نقاط المستقيم غير متساوية

الخوارزمية البسيطة

- المطلوب رسم مستقيم بين النقطتين (x0,y0), (x1,y1)
- معادلة المستقيم y=mx+b

$$m = \frac{y_1 - y_0}{x_1 - x_0} = \frac{dy}{dx}$$

$$b = y_0 - m * x_0$$

```
Void line (int x0,int y0,int x1, int y1){
    Int dx=x1-x0; int dy=y1-y0;
    Plot(x0,y0);
    If (dx !=0) {float m=(float)dy / (float)dx;
        float b=y0-m*x0;
        If (x1>x0) dx=1;
        else dx=-1;
        While (x0 != x1){
            x0=x0+dx;
            y0=round(m*x0+b);
            Plot(x0,y0);
        }}}
```

- **مساوئ الخوارزمية:**
- استخدام متحولين من النمط الأعداد الحقيقية m, b
- استدعاء التابع round
- استخدام عمليات الضرب والقسمة على الأعداد الحقيقية

طريقة المحلل التفاضلي الرقمي (DDA)
Digital Differential Analyzer

$y_{i+1}=m*(x_{i+1})+b=m*(x_i+Dx)+b=m*x_i+b+m*Dx$

$Dx=1 \rightarrow y_{i+1}=y_i+m$

تصبح المعادلات:

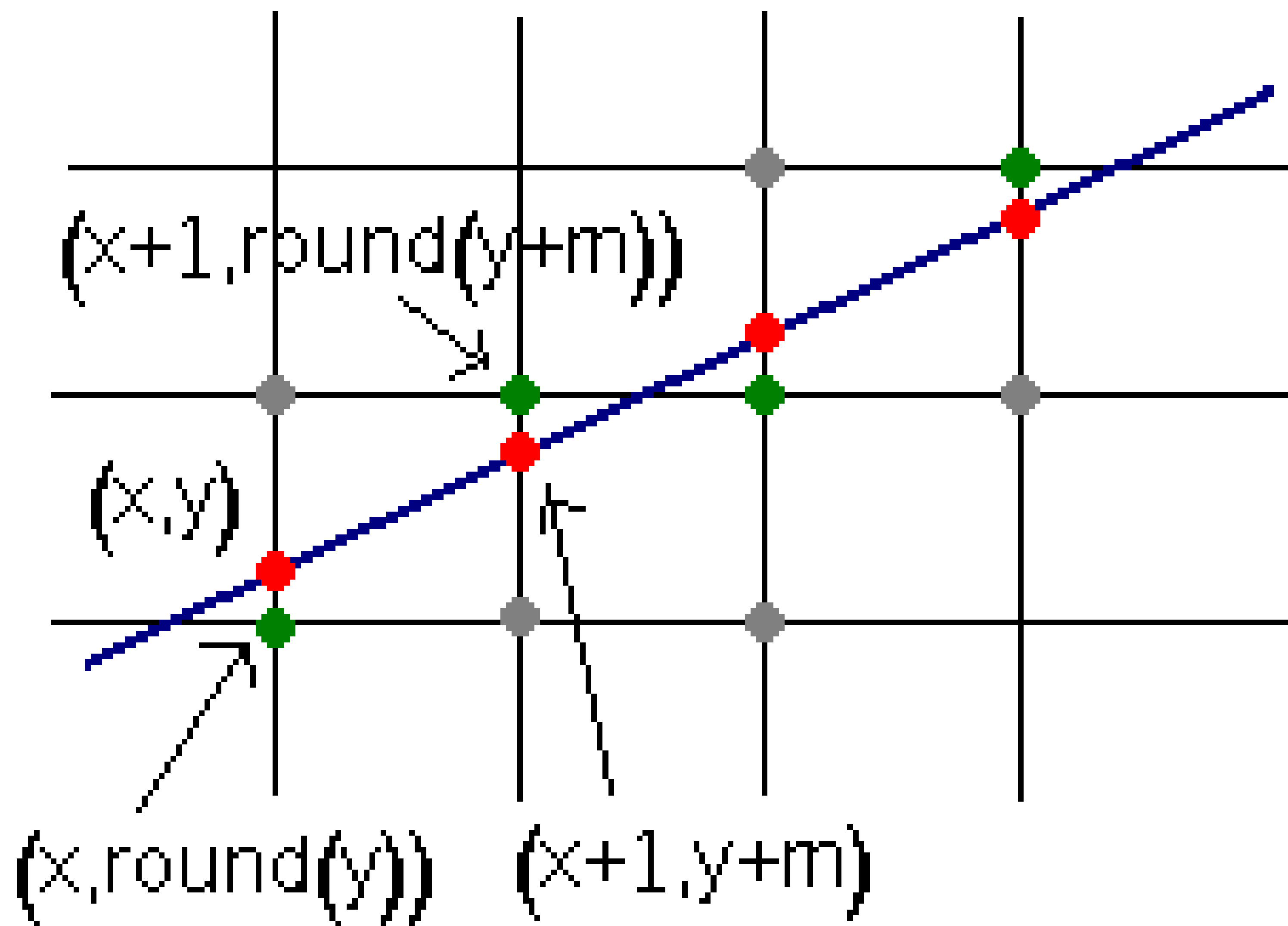
$x_{i+1}=x_i+1; \; y_{i+1}=y_i+m$

Or

$x_{i+1}=x_i-1; \; y_{i+1}=y_i-m$

الخوارزمية:

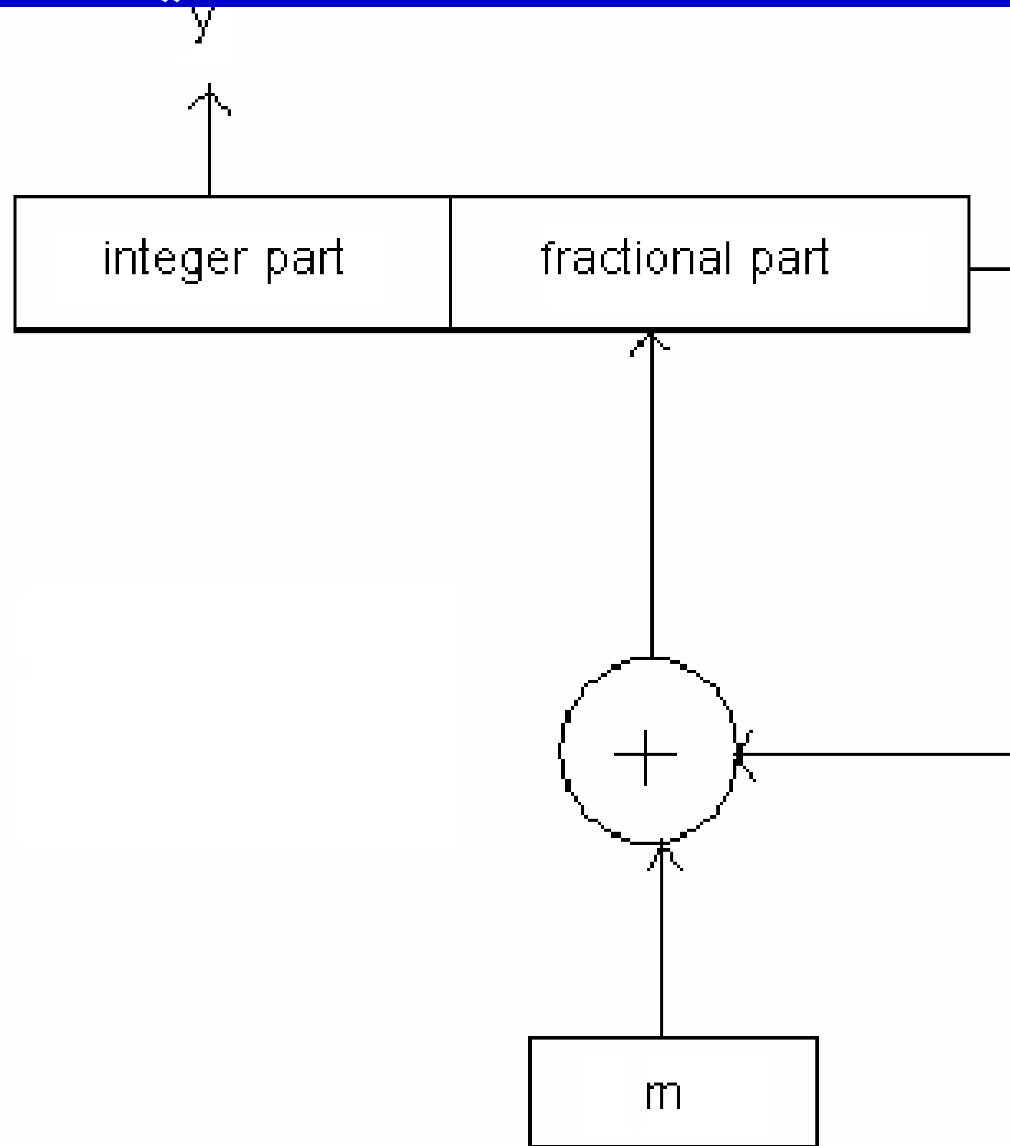
$x_{i+1}=x_i+Dx; \; mm=m*Dx \quad y_{i+1}=y_i+mm$


(x+1,round(y+m))
(x,y)
(x,round(y))
(x+1,y+m)

DDA(Digital Differential Analyzer)

```
Void DDA(int x0,int y0,int x1,int y1){
    int x;
    int dx=x1-x0;
    int dy=y1-y0;
    Double m=dy/dx;
    Double y=y0;
    If (x1>x0) Dx=1;
        else {Dx=-1;m=-1*m;}
    x=x0;
    While (x!=x1){
    Plot(x,round(y));
    x=x+Dx
    y=y+m; }}
```

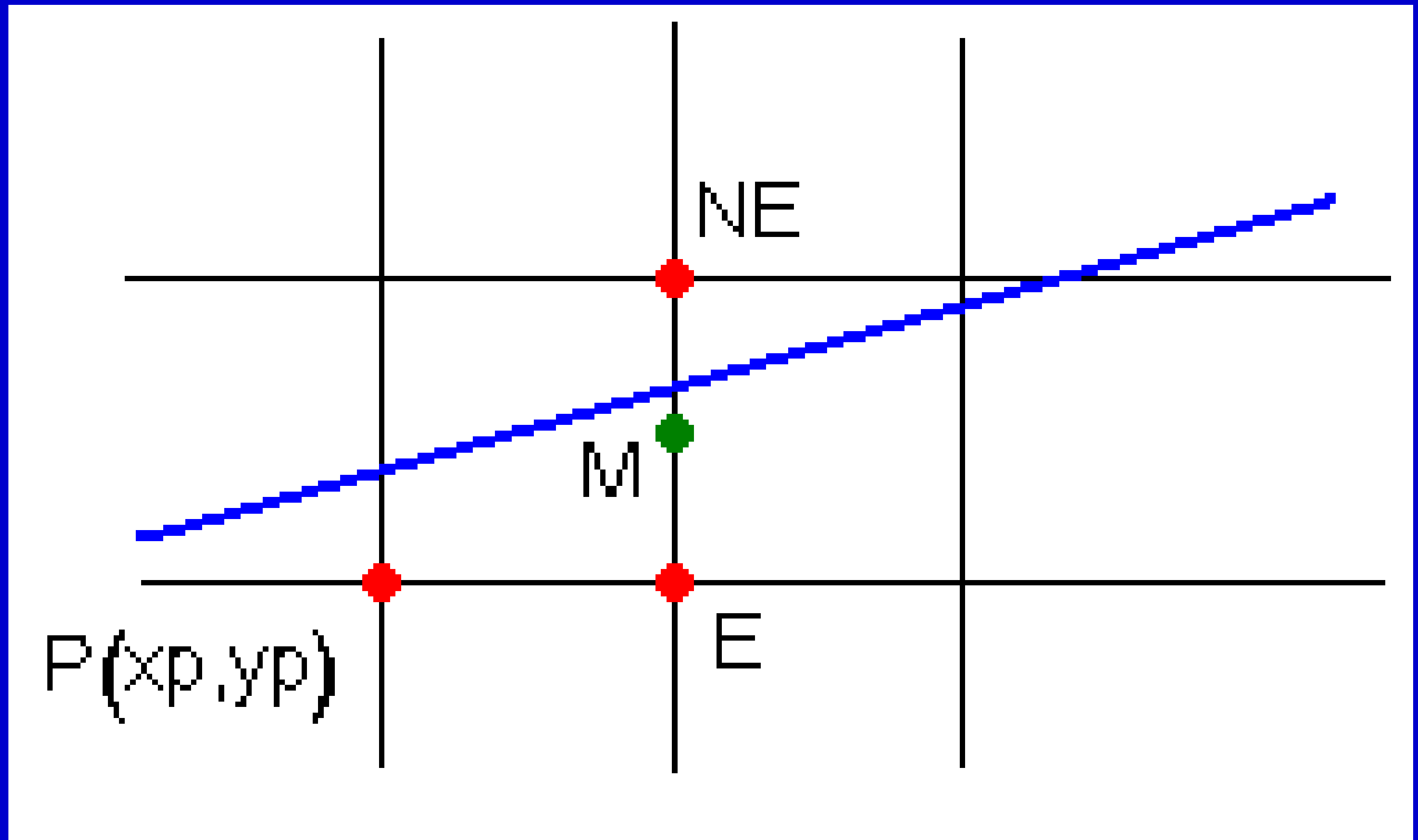
طريقة المحلل التفاضلي



- مثال ليكن p0 = (1 , 3) , p1 = (8 , 9) بالتالي نزيد x بمقدار 1 نزيد y بمقدار m

step	y	pixel
1	21/7	(1,3)
2	27/7	(2,4)
3	33/7	(3,5)
4	39/7	(4,6)
5	45/7	(5,6)
6	51/7	(6,7)
7	57/7	(7,8)
8	63/7	(8,9)

Brezenham خوارزمية



- نختار M منتصف المسافة بين E,NE
- 1- المستقيم فوق M نختار NE
- 2- المستقيم تحت M نختار E
- أكبر قيمة للخطأ=0.5
- لتحديد موقع M بالنسبة للمستقيم نستخدم معادلة المستقيم التالية:

- $F(x,y)=a.x+b.y+c=0$

- لتحديد a, b, c نستخدم المعادلة الديكارتية للمستقيم
- $y=m.x+bb$
- $m=dy/dx$

- $F(x,y)=dy.x - dx.y + bb.dx = 0$
- $a=dy$, $b= -dx$, $c= -(a.x0+b.y0)$
- احداثيات M $M(x_p+1, y_p+1/2)$
- سنعرف المتحول d= قيمة التابع عند النقطة M كمتحول قرار

$$d=F(M)=F(x_p+1,y_p+1/2)=a(x_p+1)+b(y_p+1/2)+c$$

إذا كان $d>0$ فإن M تقع تحت المستقيم لذلك نختار النقطة NE

- إذا كان $d<0$ فإن M تقع فوق المستقيم لذلك نختار النقطة E

- إذا كان d=0 فإن M تقع على المستقيم لذلك نختار النقطة E أو NE (سنختار NE).
- بعد تحديد أحد النقطتين E,NE , نحدد M الجديدة وبالتالي d الجديدة , لذلك سنحاول إيجاد خطوة لمتحول القرار d حيث نضيفها له دون حسابها من المعادلات.
- $d_{old}=F(M)=F(x_p+1,y_p+1/2)=a(x_p+1)+b(y_p+1/2)+c$
- 1- إذا اخترنا E فإن:
- $d_{new}=F(M)=F(x_p+2,y_p+1/2)=a(x_p+2)+b(y_p+1/2)+c$
- $d_{new=}d_{old}+a$ → $RightInc=a=dy$

- 2- إذا اخترنا NE:
- $d_{new}=F(M)=F(x_p+2,y_p+3/2)=a(x_p+2)+b(y_p+3/2)+c$
- $d_{new=}d_{old}+a+b$ ⇒ DigonalInc=a+b=dy-dx

RightInc و DigonalInc تمكننا من حساب قيمة القرار d في الخطوة التالية دون استخدام المعادلات.

- ما هي القيمة الابتدائية لـ d؟
- $d=F(M)=F(x_0+1,y_0+1/2)=a(x_0+1)+b(y_0+1/2)+c= F(x_0,y_0)+a+b/2=0+a+b/2=dy-dx/2$

```
void bresenham ( int x0 , int y0 , int x1 , int y1 ) {
int a = y1 – y 0 ;
int b = x0 – x1 ;
int d = 2 * a + b;
int diagonalInc = 2 * ( a + b ) ;
int rightInc = 2 *a;
int x = x0;
int y = y0;
While (x<=x1){
    Plot(x,y);
    If (d>=0){//move diagonally
    y+=1;
    d+=diagonalInc;
    }
    else { d+=rightInc;}
    x=x+1;
    }
}
```

ميزات هذه الخوارزمية:

- تتعامل مع قيم صحيحة
- لا تستخدم عمليات الضرب والقسمة وعملية round
- توجد دارة الالكترونية تحقق خوارزمية برزنهام
- الخطأ الأعظمي 0.5